

Serial No. 09/801,836
Amdt. dated August 2, 2004
Reply to Office Action of March 12, 2004

Docket No. P-0198

REMARKS/ARGUMENTS

Favorable reconsideration of this application as presently amended and in light of the following discussion is respectfully requested.

Claims 1-7, 9-15 and 17 are pending in the present application. Claims 1-7, 9-15 and 17 have been amended and claims 8 and 16 have been canceled by the present amendment.

In the outstanding Office Action, a Substitute Specification was requested; and claims 1-17 were rejected under 35 U.S.C. § 103(a) as unpatentable over Chang et al. in view of Friedlander et al. and Lehtinen.

As requested, enclosed is a Substitute Specification. It is believed no new matter has been added. Also enclosed is a marked up copy showing the changes made to the specification.

Claims 1-17 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Chang et al. in view of Fridlander et al. and Lehtinen. This rejection is respectfully traversed.

Independent claim 1 is directed to a method a method of transmitting an INAP (Intelligent Network Application Protocol) in an IN (Intellectual Network). The method includes generating an INAP message object indicating an INAP initial setting through an INAP factory object, obtaining an invoke ID and dialogue ID from a TCAP (Transaction Capabilities Application Part) dialogue object, to set them in the INAP message object, issuing a component addition command, to add the INAP message object to a TCAP

message object and generating and executing different transmission TCAP events based on a dialogue state, to send the INAP message and delete the object after sending the message.

With reference to Figure 9, the call processing object obtains the invoke ID and dialogue ID from the TCAP dialogue objects to set them in the INAP message to send the generated INAP object 46-1 to the TCAP (S4 and 85). Here, since the INAP message is sent through the TCAP dialogue object 31, the INAP object 46-1 of the INAP message block 40 calls a transmission component function of the TCAP dialogue object 31 using the INAP object itself as a parameter (S6). The TCAP dialogue object 31 gives a component command to the TCAP message object 32 included therein, to add the INAP object 46-1 to the TCAP message object 32 (S7). The TCAP dialogue object 31 calls the transmission component function to send the INAP object 46-1 added to the TCAP message object 32 (S8), to generate a transmission TCAP event that is a TCAP dialogue event (S9). In addition, it executes the transmission TCAP event to send the INAP message (S10), and then performs deletion (S11) (see paragraph [0049]).

Thus, the INAP processing method for communication between the SSP and TCAP according to the present invention allows various applications using the INAP in the SSP to be able to freely communicate with the TCAP. Furthermore, the software blocks for processing the INAP is object-oriented-designed and realized to improve reutilization thereof so that only INAP messages are added, changed and deleted without changing the

applications using the software blocks to be applied to a new system and, simultaneously, the maintenance becomes simplified (see paragraph [0069]).

The Office Action recognizes Chang et al. does not teach or suggest adding the invoke ID and dialogue ID and relies on Friedlander et al. as teaching this feature and cites col. 7, line 5 to col. 8, line 67. However, it is respectfully noted this section of Friedlander et al. merely teaches that a typical TCAP definition includes a dialogue identifier (dialogue ID), which maintains a multi-message exchange dialogue between two components (see col. 8, lines 44-47). However, there is no description in this section in which the invoke ID and dialogue from a TCAP dialogue object are set in an INAP message object.

Further, independent claim 5 has been amended to include the subject matter recited in dependent claim 8. That is, independent claim 5 now recites a method in which executing the TC primitive to which the TC beginning object was added includes, when only the TC primitive is received, executing the TC primitive, when a TC component is received together with the TC primitive, processing the TC component while withholding the execution of the TC primitive, and carrying out the withheld TC primitive, and then executing the processed TC component.

Regarding the subject matter recited in claim 8, the Office Action indicates the order of operation and execution regarding the processing of TC primitives and components received simultaneously is merely a design choice or preference for one of ordinary skill in the art at the time the invention was made. However, it is respectfully noted that independent claim 5

Serial No. 09/801,836
Amdt. dated August 2, 2004
Reply to Office Action of March 12, 2004

Docket No. P-0198

specifically recites that when only the TC primitive is received without accompanying a TC component, the TC primitive is executed to which the TC beginning object was added, and that when only the TC primitive is received, the TC primitive is executed, and when a TC component is received together with the TC primitive, the TC component is processed while withholding the execution of the TC primitive.

That is, the feature of executing a TC primitive when it is received without the TC component, and processing the TC component first while withholding the execution of the TC primitive when both the component and primitive are received, is not merely a design choice. These features are discussed in detail in paragraphs [0062] - [0065] and in Figures 14 and 15. Similar arguments apply to independent claim 14, which has been amended to include the subject matter recited in dependent claim 16. That is, the subject matter now recited in independent claim 14 is not merely a design choice nor the continuation of back-and-forth signaling inherent in SS7. It is respectfully submitted none of the applied references teach or suggest the specific features recited in the claims or the combinations thereof.

Accordingly, it is respectfully submitted independent claims 1, 5 and 14 and each of the claims depending therefrom are also allowable.

In addition, the abstract has been amended to correct minor informalities. It is believed no new matter has been added.

Serial No. 09/801,836
Amdt. dated August 2, 2004
Reply to Office Action of March 12, 2004

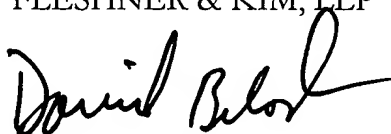
Docket No. P-0198

CONCLUSION

In view of the foregoing amendments and remarks, it is respectfully submitted that the application is in condition for allowance. Favorable consideration and prompt allowance are earnestly solicited. If the Examiner believes that any additional changes would place the application in better condition for allowance, the Examiner is invited to contact the undersigned attorney, David A. Bilodeau, at the telephone number listed below.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this, concurrent and future replies, including extension of time fees, to Deposit Account 16-0607 and please credit any excess fees to such deposit account.

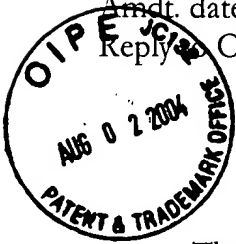
Respectfully submitted,
FLESHNER & KIM, LLP



Daniel Y.J. Kim, Esq.
Registration No. 36,186
David A. Bilodeau, Esq.
Registration No. 42,325

P.O. Box 221200
Chantilly, Virginia 20153-1200
703 766-3701 DYK/DAB:knv:dq
Date: August 2, 2004

Documents\2000\2000-471\35950.doc



ABSTRACT OF THE DISCLOSURE

~~There is provided an~~ An INAP processing method for communication between [[an]] a SSP and TCAP. When an INAP message is sent from the INAP of the SSP to the TCAP, a TC component to be transmitted is generated, an invoke ID and dialogue ID are allocated to the generated TC component, and a TC primitive corresponding to the state of an allocated dialogue is created and encoded, to be sent to the TCAP through the TCAP interface block. ~~In case where~~ When the INAP of [[an]] the SSP receives an INAP message from the TCAP, the received INAP message is decoded based on the type thereof and a corresponding SCP relationship object is found using the dialogue ID included in the decoded INAP message. When the IC component and IC primitive are contained in the decoded message, the IC primitive is executed first and then the IC component is processed. When a corresponding call has been processed, the found SCP relationship object is deleted, finishing the dialogue with the SCP. Accordingly, various applications in the SSP can freely communicate with the TCAP using the INAP message.